UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/606,641	06/29/2000	Kimberly J. Rush	MICR0512	7703

27792   7590   01/03/2006

RONALD M. ANDERSON
MICROSOFT CORPORATION
600 108TH AVENUE N.E., SUITE 507
BELLEVUE, WA 98004

EXAMINER
CHOUDHURY, AZIZUL Q

ART UNIT	PAPER NUMBER
2145	

DATE MAILED: 01/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	09/606,641	RUSH ET AL.
	Examiner	Art Unit	
	Azizul Choudhury	2145	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>30 September 2005</u>.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>21-50</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>21-50</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>29 June 2000</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

Detailed Action

This office action is in response to the correspondence received on September 30,

2005.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

Claims 21-50 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Doyle et al (US Pat No: 5838906) in view of Held et al (US Pat No: 5802367),

hereafter referred to as Doyle and Held, respectively.

1. With regards to claim 21, Doyle teaches through Held, the method for accessing

   multiple types of electronic content, comprising: receiving a request for a

   computer program to process an input to obtain an output comprising a type of

   content that is unknown to the computer program, wherein a service manager

   connects the computer program to at least one service container to process the

   input to obtain the output, wherein said at least one service container includes at

   least a data object, a code object, and a loader identification that enable

   definition of the type of content unknown to the computer program; selecting at

least one segment of computer code from a plurality of segments of computer

code that will enable the computer program to process the input, when the at

least one segment of computer code is executed along with the computer

program, to provide the output comprising the type of content that is unknown to

the computer program, the at least one segment of computer code being

selected without reference to any embedded link by said computer program; and

executing the at least one segment of computer code along with the computer

program to process the input and obtain the output comprising the type of

content that is unknown to the computer program, wherein the plurality of

segments of computer code and the at least one segment of computer code are

not executable as an independent computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link. Multiple surrogate process means are taught by Held (column 10, line 14 – column 20, line 37, Held) however, in each of the means, an instance of the surrogate process is equivalent to the service container. Within an instance of a surrogate process, there can be found server code, a registration database and objects. For example purposes, correlations between these surrogate process elements and the claimed service container elements will be drawn using a surrogate process in a network environment (column 10, line 14 – column 11, line 40, Held). An instance of a surrogate process requires objects and multiple types of objects are available that are equivalent to the claimed data object, including class factory object, new object or existing object. As for the claimed code object, Held's design makes use of something called "server code." A "server code" is used to reference to executable code. Finally, Held's design makes use of a registration database. It allows elements to be located and hence means for the claimed "loader identification" are present within Held's design as well.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

2. With regards to claim 22, Doyle teaches through Held, a method wherein selecting

at least one segment of computer code comprises selecting at least two

segments of computer code from the plurality of segments of computer code

whose combined functionality will enable the computer program to process the

input, when the at least two segments of computer code are executed along with

the computer program, to provide the output comprising the type of content that

is unknown to the computer program, the at least two segments of computer

code being selected based upon the functionality that each provides

    (Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is

given as to how many applications may be selected to run together, hence two or

more may be selected as claimed. In addition, since the list of applications is

initiated by the system, there must exist a service manager as claimed.

However, such means are not expressly stated within Doyle's design.

    Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held teaches how processes are selected based on how well it will function (for instance with regards to compatibility with the first program) (column 7, line 58 – column 8, line 12, Held).

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

3. With regards to claim 23, Doyle teaches through Held, the method further comprising configuring the at least two segments of computer code to be executed along with the computer program in a particular order to provide a desired processing of the input

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide, as claimed. No limitation is given as to how many applications may be selected to run together; hence two or more may be selected as claimed. Whenever multiple applications run together to produce a single output, it is inherent that they will perform in order as claimed.

In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

4. With regards to claim 24, Doyle teaches through Held, a method further comprising configuring the at least two segments of computer code into a master-slave relationship that causes the execution of one of the at least two segments of computer code to be dependent on the execution of another of the at least two segments of computer code

(Doyle discloses how applications can be started as child processes (column 15, line 22, Doyle). When a child process exists, a parent process must exist. This parent-child relationship is equivalent to a master-slave relationship. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

5. With regards to claim 25, Doyle teaches through Held, a method wherein executing the at least one segment of computer code comprises integrating the at least one segment of computer code into the computer program and executing

the computer program to process the input and obtain the output comprising the type of content that is unknown to the computer program

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). This list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide. When an application runs along with the original program, Doyle states that it is "embedded" (column 6,, lines 58-59, Doyle). Embedded is equivalent to the claimed integrated. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for

transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

6. With regards to claim 26, Doyle teaches through Held, a method wherein receiving

a request for a computer program to process an input comprises receiving a

command to translate a word from a first language to a second language

(Doyle discloses a design that features a "list of applications," (column 15,

line 14, Doyle). These applications are able to run along with an original program

to provide an output from an input the original program is unable to produce

alone. No limitation is placed on the type of programs available, hence

translation programs are usable within the scope of the design. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, no limitation is placed

on the type of functionality the second (surrogate) program or process can

provide.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

7. With regards to claim 27, Doyle teaches through Held, a method wherein receiving

a request for a computer program to process an input comprises receiving a

command to convert a number from a first number format to a second number

format

(Doyle discloses a design that features a "list of applications," (column 15,

line 14, Doyle). These applications are able to run along with an original program

to provide an output from an input the original program is unable to produce

alone. No limitation is placed on the type of programs available, hence number

conversion programs are usable within the scope of the design. In addition,

since the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, no limitation is placed

on the type of functionality the second (surrogate) program or process can

provide.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

8. With regards to claim 28, Doyle teaches through Held, the method wherein

receiving a request for a computer program to process an input comprises

receiving a command to convert a text object from a first text format to a second

text format

(Doyle discloses a design that features a "list of applications," (column 15,

line 14, Doyle). These applications are able to run along with an original program

to provide an output from an input the original program is unable to produce

alone. No limitation is placed on the type of programs available; hence text

format conversion programs are usable within the scope of the design. In

addition, since the list of applications is initiated by the system, there must exist a

service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link. Plus, no limitation is placed on the type of functionality the second (surrogate) program or process can provide.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

9. With regards to claim 29, Doyle teaches through Held, a method wherein receiving a request for a computer program to process an input comprises receiving a command to convert a graphical object from a first graphical format to a second graphical format

(Doyle discloses a design that features a "list of applications," (column 15, line 14, Doyle). These applications are able to run along with an original program to provide an output from an input the original program is unable to produce alone. No limitation is placed on the type of programs available; hence graphical format conversion programs are usable within the scope of the design. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link. Plus, no limitation is placed on the type of functionality the second (surrogate) program or process can provide.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

10. With regards to claim 30, Doyle teaches through Held, a computer system for accessing multiple types of electronic content, comprising: a processing unit; a memory in communication with the processing unit; and a computer program stored in the memory that provides instructions to the processing unit, wherein the processing unit is responsive to the instructions, operable for: identifying a plurality of segments of computer code that can be executed along with the computer program by the processing unit in response to the instructions; selecting, in response to an input command to access at least one type of content that the computer program is not configured to access, at least one segment of computer code from the plurality of segments of computer code that can be executed along with the computer program by the processing unit, in response to the instructions, to access the at least one type of content that the computer program is not configured to access, wherein a service manager connects the computer program to at least one service container in response to the input command, wherein said computer program is not directed to the plurality of segments of computer code by any embedded link referenced by said computer program and wherein said at least one service container includes at least a data object, a code object, and a loader identification that enable definition of the type of content unknown to the computer program; and executing the at least one segment of computer code along with the computer program to access the at least one type of content that the computer program is not

configured to access; wherein the plurality of segments of computer code and the at least one segment of computer code are not executable as an independent computer program

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide as claimed. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link. Multiple surrogate process means are taught by Held (column 10, line 14 – column 20, line 37, Held) however, in each of the means, an instance of the surrogate process is equivalent to the service container. Within an instance of a surrogate process, there can be found server code, a registration database and objects. For example purposes, correlations between these surrogate process elements and the claimed service container elements will be drawn using a surrogate process

in a network environment (column 10, line 14 – column 11, line 40, Held). An instance of a surrogate process requires objects and multiple types of objects are available that are equivalent to the claimed data object, including class factory object, new object or existing object. As for the claimed code object, Held's design makes use of something called "server code." A "server code" is used to reference to executable code. Finally, Held's design makes use of a registration database. It allows elements to be located and hence means for the claimed "loader identification" are present within Held's design as well.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

11. With regards to claim 31, Doyle teaches through Held a computer system wherein the processing unit, responsive to the instructions, is further operable for: arranging in the memory the at least one segment of computer code and a data, comprising the at least one type of content that the computer program is not configured to access, into a function-content group; and interfacing the function-content group to the computer program to enable the computer program

to access the at least one type of content that the computer program is not

configured to access

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitations

are placed as to the number of programs that may be run together. In addition,

since the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

12. With regards to claim 32, Doyle teaches through Held, the computer system

wherein the processing unit, responsive to the instructions, is operable for

identifying a plurality of segments of computer code by: locating at least two

segments of computer code from the plurality of segments of computer code that

each comprise a portion of computer code that indicates they can be executed by

the processing unit along with the computer program; and generating a list in the

memory comprising an identifier for each of the at least two segments of

computer code that indicates that the at least two segments of computer code

are available to be executed by the processing unit along with the computer

program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected as claimed. In addition, since the list of applications is

initiated by the system, there must exist a service manager as claimed.

However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held discloses that

multiple classes can be implemented by the same module (hence multiple

behaviors/function/services can be made available and selected) (column 5, lines

30-49, Held).

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

13. With regards to claim 33, Doyle teaches through Held, a computer system

wherein the processing unit, responsive to the instructions, is operable for

selecting at least one segment of computer code by selecting at least two

segments of computer code from the plurality of segments of computer code

whose combined functionality will allow the computer program to access the at

least one type of content that the computer program is not configured to access

when the at least two segments of computer code are executed by the

processing unit along with the computer program

(Doyle teaches a design that features a "list of applications," (column 15,
line 14, Doyle). The list of applications allows an application to be selected to run
along with the original program to provide an output from an input that the
original program alone would be unable to provide, as claimed. No limitation is
given as to how many applications may be selected to run together, hence two or
more may be selected as claimed. In addition, since the list of applications is
initiated by the system, there must exist a service manager as claimed.
However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be
called to assist a first program to perform a desired task on data (column 3, line
65 – column 4, line 57, Held). Within the disclosure, Held states the existence of
a service control manager (equivalent to the service manager) (column 4, line 16,
Held). In addition, Held teaches how a surrogate process can be called from a
first program without reference to an embedded link. Plus, Held discloses that
multiple classes can be implemented by the same module (hence multiple
behaviors/function/services can be made available and selected) (column 5, lines
30-49, Held).

Both Doyle and Held teach systems allowing one program to call upon
another program to perform an operation on a piece of data. Hence, it would
have been obvious to one skilled in the art, during the time of the invention, to
have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

14. With regards to claim 34, Doyle teaches through Held, the computer system

wherein the processing unit, responsive to the instructions, is further operable for

configuring the at least two segments of computer code to be executed by the

processing unit along with the computer program in a particular order to allow the

computer program to access the at least one type of content that the computer

program is not configured to access

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is

given as to how many applications may be selected to run together, hence two or

more may be selected as claimed. Whenever multiple applications run together

to produce a single output, it is inherent that they will perform in order as claimed.

In addition, since the list of applications is initiated by the system, there must

exist a service manager as claimed. However, such means are not expressly

stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held discloses that

multiple classes can be implemented by the same module (hence multiple

behaviors/function/services can be made available and selected) (column 5, lines

30-49, Held).

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

15. With regards to claim 35, Doyle teaches through Held, the computer system,

wherein the processing unit, responsive to the instructions, is further operable for

configuring the at least two segments of computer code into a master-slave

relationship that causes the execution of one of the at least two segments of

computer code to be dependent on the execution of another of the at least two

segments of computer code

(Doyle discloses how applications can be started as child processes

(column 15, line 22, Doyle). When a child process exists, a parent process must

exist. This parent-child relationship is equivalent to a master-slave relationship.

In addition, since the list of applications is initiated by the system, there must

exist a service manager as claimed. However, such means are not expressly

stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

16. With regards to claim 36, Doyle teaches through Held, a computer-readable

medium having computer-executable instructions for accessing multiple types of

electronic content, comprising: logic for creating a list that comprises information

about a plurality of segments of computer code that can be executed along with a

computer program; logic for choosing at least one segment of computer code

from the plurality of segments of computer code, based on the information in the

list, that can be executed along with the computer program to process a type of

data that the computer program is not designed to process, wherein said

computer program is not directed to the at least one segment of computer code

by any embedded link referenced by said computer program; logic to execute the

at least one segment of computer code along with the computer program in

response to an input to provide an output of the type of data that the computer

program is not designed to process, wherein a service manager connects the

computer program to at least one service container in response to the input,

wherein said at least one service container includes at least a data object, a code

object, and a loader identification that enable definition of the type of content

unknown to the computer program; and wherein the plurality of segments of

computer code and the at least one segment of computer code are not

executable as an independent computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Multiple surrogate process

means are taught by Held (column 10, line 14 – column 20, line 37, Held)

however, in each of the means, an instance of the surrogate process is

equivalent to the service container. Within an instance of a surrogate process,

there can be found server code, a registration database and objects. For

example purposes, correlations between these surrogate process elements and

the claimed service container elements will be drawn using a surrogate process

in a network environment (column 10, line 14 – column 11, line 40, Held). An

instance of a surrogate process requires objects and multiple types of objects are

available that are equivalent to the claimed data object, including class factory

object, new object or existing object. As for the claimed code object, Held's

design makes use of something called "server code." A "server code" is used to

reference to executable code. Finally, Held's design makes use of a registration

database. It allows elements to be located and hence means for the claimed

"loader identification" are present within Held's design as well.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

17. With regards to claim 37, Doyle teaches through Held, he computer-readable

medium further comprising logic for choosing at least two segments of computer

code from the plurality of segments of computer code, based on the information

in the list, which can be executed along with the computer program to process a

type of data that the computer program is not designed to process

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected claimed. In addition, since the list of applications is

initiated by the system, there must exist a service manager as claimed.

However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held discloses that

multiple classes can be implemented by the same module (hence multiple

behaviors/function/services can be made available and selected) (column 5, lines

30-49, Held).

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

18. With regards to claim 38, Doyle teaches through Held, the computer-readable

medium, further comprising logic for linking the at least two segments of

computer code in a specific order of execution to provide a desired output of data

that the computer program is not designed to process when the at least two

segments of computer code are executed along with the computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected claimed. Whenever multiple applications run together to

produce a single output, it is inherent that they will perform in order as claimed.

In addition, since the list of applications is initiated by the system, there must

exist a service manager as claimed. However, such means are not expressly

stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held discloses that

multiple classes can be implemented by the same module (hence multiple

behaviors/function/services can be made available and selected) (column 5, lines

30-49, Held).

Both Doyle and Held teach systems allowing one program to call

upon another program to perform an operation on a piece of data. Hence, it

would have been obvious to one skilled in the art, during the time of the

invention, to have combined the teachings of Doyle with those of Held, to provide

a method for transparently executing code using a surrogate process (column 3,

line 67 – column 4, line 1, Held)).

19. With regards to claim 39, Doyle teaches through Held, the computer-readable

medium, wherein the logic for creating a list that comprises information about a

plurality of segments of computer code comprises: logic for identifying at least

two segments of computer code that each comprise a registration code that

indicates that they can be executed along with the computer program; and logic

for generating a list comprising an identification code for each of the at least two

segments of computer code that indicates that the at least two segments of

computer code are available to be executed along with the computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected claimed. Whenever multiple applications run together to

produce a single output, it is inherent that they will perform in order as claimed.

Furthermore, Doyle describes a design that has the means by which to identify

the application to be launched from the local user client machine (column 15,

lines 18-21, Doyle). In addition, since the list of applications is initiated by the

system, there must exist a service manager as claimed. However, such means

are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held discloses that

multiple classes can be implemented by the same module (hence multiple

behaviors/function/services can be made available and selected) (column 5, lines

30-49, Held).

Both Doyle and Held teach systems allowing one program to call

upon another program to perform an operation on a piece of data. Hence, it

would have been obvious to one skilled in the art, during the time of the

invention, to have combined the teachings of Doyle with those of Held, to provide

a method for transparently executing code using a surrogate process (column 3,

line 67 – column 4, line 1, Held)).

20. With regards to claim 40, Doyle teaches through Held, computer-readable

medium, further comprising logic for arranging the at least one segment of

computer code and a data element, comprising the type of data that the

computer program is not designed to process, into a function-data group; and

logic for interfacing the function-data group to the computer program to enable

the computer program to provide the output of the type of data that the computer

program is not designed to process

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or more may be selected claimed. Whenever multiple applications run together to produce a single output, it is inherent that they will perform in order as claimed. Furthermore, for data to be processed, it typically is grouped together, for instance within a data structure of some form. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Held, to provide a method for transparently executing code using a surrogate process (column 3, line 67 – column 4, line 1, Held)).

21. With regards to claims 41, 44 and 47, Doyle teaches through Held, the method

wherein each service container comprises a data object, a code object, and a

loader identification

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager along with reference pointers as claimed. However, such means are

not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Multiple surrogate process

means are taught by Held (column 10, line 14 – column 20, line 37, Held)

however, in each of the means, an instance of the surrogate process is

equivalent to the service container. Within an instance of a surrogate process,

there can be found server code, a registration database and objects. For

example purposes, correlations between these surrogate process elements and

the claimed service container elements will be drawn using a surrogate process

in a network environment (column 10, line 14 – column 11, line 40, Held). An

instance of a surrogate process requires objects and multiple types of objects are

available that are equivalent to the claimed data object, including class factory

object, new object or existing object. As for the claimed code object, Held's

design makes use of something called "server code." A "server code" is used to

reference to executable code. Finally, Held's design makes use of a registration

database. It allows elements to be located and hence means for the claimed

"loader identification" are present within Held's design as well.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

22. With regards to claims 42, 45 and 48, Doyle teaches through Held, the method

wherein each code object references at least one service object

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager along with reference pointers as claimed. However, such means are not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link. Multiple surrogate process means are taught by Held (column 10, line 14 – column 20, line 37, Held) however, in each of the means, an instance of the surrogate process is equivalent to the service container. Within an instance of a surrogate process, there can be found server code, a registration database and objects. For example purposes, correlations between these surrogate process elements and the claimed service container elements will be drawn using a surrogate process in a network environment (column 10, line 14 – column 11, line 40, Held). An instance of a surrogate process requires objects and multiple types of objects are available that are equivalent to the claimed data object, including class factory object, new object or existing object. As for the claimed code object, Held's design makes use of something called "server code." A "server code" is used to reference to executable code. Finally, Held's design makes use of a registration database. It allows elements to be located and hence means for the claimed "loader identification" are present within Held's design as well.

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

23. With regards to claims 43, 46 and 49, Doyle teaches through Held, a method

wherein each service object is stored in a cache, separate from each service

container

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager along with reference pointers as claimed. However, such means are

not expressly stated within Doyle's design.

Held discloses a design allowing a second program or process to be

called to assist a first program to perform a desired task on data (column 3, line

65 – column 4, line 57, Held). Within the disclosure, Held states the existence of

a service control manager (equivalent to the service manager) (column 4, line 16,

Held). In addition, Held teaches how a surrogate process can be called from a

first program without reference to an embedded link. Plus, Held discloses that

means for caching exist within the design (column 16, line 62 – column 17, line 5,

Held).

Both Doyle and Held teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

24. With regards to claim 50, Doyle teaches through Held, a method for accessing;

multiple types of electronic content from a program, comprising the steps of:

defining a plurality of service containers accessible to the program, each service

container corresponding to a specific function, each service container including:

a data object, such that each data object includes data required to enable the

specific function corresponding to the service container to be achieved; a code

object, each code object referencing at least one segment of programming code

stored separately from the service container, such that the at least one segment

of programming code referenced by the code object includes programming code

required to enable the specific function corresponding to the service container to

be achieved; and a loader identification, the loader identification providing the

program with information required to load the segment of programming code

referenced by the code object; requesting an input to be processed to obtain an

output; identifying one of the plurality of service containers whose corresponding

specific function is able to process the input; parsing the service container thus

identified, to determine: data required to enable the specific function

corresponding to the service container to be achieved; an identity of the at least

one segment of programming code required to enable the specific function

corresponding to the service container to be achieved; and information required

to load each segment of programming code referenced by the code object;

retrieving the at least one segment of programming code required to enable the

specific function corresponding to the service, container to be achieved; and

executing the at least one segment of computer code under the control of the

computer program to process the input and obtain the output, wherein said

computer program is not directed to the at least one segment of computer code

by any embedded link referenced by said computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Held discloses a design allowing a second program or process to be called to assist a first program to perform a desired task on data (column 3, line 65 – column 4, line 57, Held). Within the disclosure, Held states the existence of a service control manager (equivalent to the service manager) (column 4, line 16, Held). In addition, Held teaches how a surrogate process can be called from a first program without reference to an embedded link. Multiple surrogate process means are taught by Held (column 10, line 14 – column 20, line 37, Held) however, in each of the means, an instance of the surrogate process is equivalent to the service container. Within an instance of a surrogate process, there can be found server code, a registration database and objects. For example purposes, correlations between these surrogate process elements and the claimed service container elements will be drawn using a surrogate process in a network environment (column 10, line 14 – column 11, line 40, Held). An instance of a surrogate process requires objects and multiple types of objects are available that are equivalent to the claimed data object, including class factory object, new object or existing object. As for the claimed code object, Held's design makes use of something called "server code." A "server code" is used to reference to executable code. Finally, Held's design makes use of a registration database. It allows elements to be located and hence means for the claimed "loader identification" are present within Held's design as well.

Both Doyle and Held teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Held, to provide a method

for transparently executing code using a surrogate process (column 3, line 67 –

column 4, line 1, Held)).

Response to Remarks

The amendment received on September 30, 2005 has been carefully examined

but is not deemed fully persuasive. The claim amendments have been reviewed and

the prior art has been further evaluated. After extensive analysis of the prior art, the

examiner has concluded that multiple surrogate process means are taught by Held

(column 10, line 14 – column 20, line 37, Held) however, in each of the means, an

instance of the surrogate process is equivalent to the service container. Multiple

instances of surrogate processes are possible. Within an instance of a surrogate

process, there can be found server code, a registration database and objects. For

example purposes, correlations between these surrogate process elements and the

claimed service container elements will be drawn using a surrogate process in a

network environment (column 10, line 14 – column 11, line 40, Held). An instance of a

surrogate process requires objects and multiple types of objects are available that are

equivalent to the claimed data object, including class factory object, new object or

existing object. As for the claimed code object, Held's design makes use of something

called "server code." A "server code" is used to reference to executable code. Finally,

Held's design makes use of a registration database. It allows elements to be located

and hence means for the claimed "loader identification" are present within Held's design

as well.

Conclusion

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Azizul Choudhury whose telephone number is (571)

272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Jason Cardone can be reached on (571) 272-3933. The fax phone number

for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

AC

ZARNI MAUNG
SUPERVISORY PATENT EXAMINER